UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P O Box 1450
Alexandria, Virginia 22313-1450
www uspto gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/714,730	11/17/2003	King-Hwa Lee	OIC0078US	4654

60975 7590 02/03/2009
CAMPBELL STEPHENSON LLP
11401 CENTURY OAKS TERRACE
BLDG. H, SUITE 250
AUSTIN, TX 78758

EXAMINER	
TANK, ANDREW L	
ART UNIT	PAPER NUMBER
2175	
MAIL DATE	DELIVERY MODE
02/03/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	**Application No.** 10/714,730	**Applicant(s)** LEE ET AL.	
	Examiner Andrew Tank	**Art Unit** 2175	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *15 October 2008*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-8,10-20 and 22-31* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-8,10-20 and 22-31* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application

6) ☐ Other: _____.

DETAILED ACTION

1. The following action is in response to the amendment filed October 15, 2008. Claims 1, 8, 11, 20 and 22-24 have been directly amended. **Claims 1-8, 10-20, and 22-31** are pending and have been considered below.

Claim Rejections - 35 USC § 112

2. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

3. **Claims 1-8 and 10-19** are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claims 1-8 and 10-19: Claim 1 recites the amended limitation: "the user interface comprises a configurable subset of the data" in line 16. However there is both a “data structure” in line 8 and a “data element" in line 13. It is unclear as to which data, the data structure or the data element, the limitation is referring. The examiner will interpret "the data” to be “the data element”. Claims 8 and 11 recite similar limitations and are rejected for similar reasons.

Claim Rejections - 35 USC § 101

4. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

5. **Claims 20 and 22-24** are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

Claims 20 and 22-24: Claim 20 and 24 recite: "A computer-readable storage medium containing: instructions, executable on a computer system, configured to execute [..]". While this is normally acceptable, Applicant further adds from the specification (paragraph [0017]): "The memory and storage devices are computer-readable media that may contain instructions that implement the file sharing system. In addition, the data structures and message structures may be stored or transmitted via a data transmission medium, such as a signal on a communications link." (-Emphasis) Therefore, claims 20 and 22-24 must be given their broadest interpretation in view of the specification and are drawn to instructions stored in a propagated signal. A propagated signal is not a series of steps or acts and this is not a process. A propagated signal is not a physical article or object and as such is not a machine or manufacture. A propagated signal is not a combination of substances and therefore not a compilation of matter. Thus, a propagated signal does not fall within any of the four categories of invention. Therefore, claim 20 and 22-24 are not statutory.

Claim Rejections - 35 USC § 103

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

7. **Claims 20 and 22-23** are rejected under 35 U.S.C. 103(a) as being unpatentable over Bookman et al., (US 5,761,673), previously presented as Bookman.

Claim 20: Bookman discloses a computer-readable medium containing instructions, executable on a computer system, configured to execute a command of a business application, and a data structure defining the command, wherein the command is inbound to a web server, and the web server is configured to execute on the computer system (col 3 lines 61-62: "Web server environment, containing conventional objects"), the data structure comprising:

an execute element having a path attribute indicating a location of an object manager (col 4 lines 1-2: "Web server executable");

a command element nested within the execute element and having a value attribute indicating a name of the command (col 4 lines 2-3: "Web browser makes an object request from Web server executable"), wherein

the command element represents a predefined query (col 4 lines 5-6: "a table in database, based on the object request URL"); and

one or more argument elements nested within the command element, each argument element having a name attribute indicating a name of an argument for the named command (col 3 lines 64-65: "each object has an attribute associated with it"), the one or more argument elements being from a set of argument elements comprising an argument element for indicating a response markup format (col 4 line 7: "an HTML file"), an argument element for indicating whether the response should include user interface elements (col 4 line 7: "an HTML file"), and an arguments element identifying a transform to be applied to output (col 4 lines 9-12: "CGI script").

Bookman further discloses that the web server environment contains a multitude of computers (col 1 lines 12-14) and programs (col 1 lines 22-24). Data is supplied to and from the computers and servers (col 1 lines 44-52). Therefore it would have been obvious to one having ordinary skill in the art and the teachings of Bookman before them at the time the present invention was made to apply the known technique of a network of computers and a server and passing of information between them, taught by Bookman, improve the similar device taught by Bookman above to yield the predictable result of expanding Bookman's device above to further contain a plurality of instructions executable on a plurality of computer systems and configured to execute a plurality of commands on a plurality of business applications according to the data structures and methods of Bookman above.

Claim 22: Bookman discloses a computer-readable medium as in claim 20 above, and further disclose that zero or more occurrences of the command elements are nested within the execute element. Bookman discloses that a command element is nested within the execute element, therefore, the command elements, when they do occur (one), are nested within the execute element and, when they do not occur (zero), occur nowhere (col 3 lines 61-67, col 4 lines 1-12).

Claim 23: Bookman discloses the computer-readable medium as in claim 20 above, but do not specifically disclose that only one command element is nested within the execute element. However, Bookman does disclose a command element nested within the execute element (col 3 lines 61-67, col 4 lines 1-12). Therefore, it would have been obvious to one of ordinary skill in the art at the time the present invention was made to include only one command element within the execute element. One would have been motivated to only include one command element when only one command element was needed, in order to save processing time and increase the efficiency and speed with which the processor operates.

8. **Claims 1-8, 10-19, and 25-31** are rejected under 35 U.S.C. 103(a) as being unpatentable over London et al. (US 5,831,609), previously presented as London, in view of Hallberg et al., "Using Microsoft Excel 97", published by Que Corporation, copyright 1997 Que Corporation, previously presented as "Hallberg".

Claim 1: London discloses a method compromising:

providing information relating to a business application (col 2 lines 60-63: "host computer executes application programs") in a server system (col 3 lines 31-32: "the host computer is connected to the X-Terminals"), comprising:

receiving a request (col 3 lines 31-49 "network"), wherein

the request is configured to cause the business application to execute a command of the business application (col 5 lines 6-7: "when the API command is window management related"),

the request comprises an indication of a user interface element to be returned (col 4 lines 55-67),

generating a data element by executing the command of the business application to (col 6 lines 27 "character string Hello");

generating the user interface element to be returned in response to the request (col 6 lines 17-18 "ShowWindow API"), wherein the user interface element comprises a configurable subset of the data element (col 6 line 27 "Hello" is a character string, i.e. configurable subset of data); and

sending a response comprising user interface element and the data element (col 5 lines 65-67, col 6 line 1).

While London does not specifically disclose that the command comprises an execute, command, and argument element and further that the command element is a predefined query, London does disclose the use of "MICROSOFT EXCEL" (col 2 line 62). Hallberg discloses that Microsoft Excel is a spreadsheet program with database functionality (page 383: "Building Excel Databases"). This functionality includes the ability to selectively filter list results (page 394: "AutoFilter"), i.e. query the database ("*command element*" to return ("*execute element*") results based on a predefined filter ("*argument element*"). Therefore, it would have been obvious to one having ordinary skill in the art and the teachings of London and Hallberg before them at the time the present invention was made, to use the business application Microsoft Excel, complete with the database functionality and querying, in the business application interface server system disclosed by London. One would have been motivated to allow a user to use Microsoft Excel, in particular the database functionality of Excel, in order to better manage data, as suggested by Hallberg (page 383: "organizing your data").

London and Hallberg disclose method as above, but do not specifically disclose that the generated user interface and data elements are XML documents specified by XSLT stylesheets. However, London does disclose the use of the X-Protocol to generate an output.

One of ordinary skill in the art at the time the present invention was made would know to that what is written one programming language can also be written in another programming language such as C, C++, HTML, JavaScript, X-Protocol, XML, WML, etc. Therefore, it would have been obvious to one of ordinary skill in the art at the time the present invention was made to generate the output in XML format, and to specify the presentation of a class of XML documents using XSLT stylesheets. One would have been motivated to do this in order to have a result more compatible with web-based applications.

Claim 8: London discloses a method in a server system for providing application information, the method compromising:

providing transforms for transforming output of the business application, each transform having a name (col 7 lines 16-30);

receiving from a client system a request to execute a command of an application (col 4 lines 48-53), the request optionally indicating the name of a transform to be applied to the output of the application (col 7 lines 16-30);

generating a data element by executing the command of the business application to (col 6 lines 27 "character string Hello");

executing the command of the application to generate output (col 5 lines 22-31) from a configurable subset of the data element (col 6 lines 27 "Hello" is a character string, i.e. a configurable subset of the data element);

when the request indicates the name of a transform, generating a transformed output by applying the provided transform with the indicated name to the generated output (col 7 lines 10-30); and

otherwise, sending to the client system the generated output (col 7 lines 10-30).

While London does not specifically disclose that the command comprises an execute, command, and argument element and further that the command element is a predefined query, London does disclose the use of "MICROSOFT EXCEL" (col 2 line 62). Hallberg discloses that Microsoft Excel is a spreadsheet program with database functionality (page 383: "Building Excel Databases"). This functionality includes the ability to selectively filter list results (page 394: "AutoFilter"), i.e. query the database ("*command element*" to return ("*execute element*") results based on a predefined filter ("*argument element*"). Therefore, it would have been obvious to one having ordinary skill in the art and the teachings of London and Hallberg before them at the time the present invention was made, to use the business application Microsoft Excel, complete with the database functionality and querying, in the business application interface server system method disclosed by London. One would have been motivated to allow a user to use Microsoft Excel, in particular the database functionality of Excel, in order to better manage data, as suggested by Hallberg (page 383: "organizing your data").

London and Hallberg disclose method as above, but do not specifically disclose that the requests and outputs are XML documents specified by XSLT stylesheets. However, London does disclose the use of the X-Protocol to generate an output. One of ordinary skill in the art at the time the present invention was made would know to that what is written one programming language can also be written in another programming language such as C, C++, HTML, JavaScript, X-Protocol, XML, WML, etc. Therefore, it would have been obvious to one of ordinary skill in the art at the time the present invention was made to generate the output in XML format, and to specify the presentation of a class of XML documents using XSLT stylesheets. One would have been motivated to do this in order to have a result more compatible with web-based applications.

Claim 25: London discloses a method in a server system for providing application information, the method comprising:

receiving a request (col 3 lines 31-49 "network") to execute a command of a business application (col 4 lines 48-53), wherein

the request is received from a client system (col 5 lines 65-67, col 6 line 1),

the command indicates a user interface element (col 4 lines 53-67) and a data element (col 6 lines 21-22: "requests the painting of the text Hello on the screen") to be returned as results of execution of the command;

generating the data element by executing the command (col 6 line 27: "character string Hello");

when the command indicates to return at least one user interface element (col 5 lines 53-56), generating the at least one user interface element to be returned; and

sending a first response to the client system a response that comprises the at least one user interface element and the data element (col 5 lines 65-67, col 6 line 1); and

otherwise (col 5 lines 48-52),

sending a second response to the client system comprising the generated data element without user interface element (col 5 lines 65-67, col 6 line 1).

While London does not specifically disclose that the command comprises an execute, command, and argument element and further that the command element is a predefined query, London does disclose the use of "MICROSOFT EXCEL" (col 2 line 62). Hallberg discloses that Microsoft Excel is a spreadsheet program with database functionality (page 383: "Building Excel Databases"). This functionality includes the ability to selectively filter list results (page 394: "AutoFilter"), i.e. query the database ("*command element*" to return ("*execute element*") results based on a predefined filter ("*argument element*"). Therefore, it would have been obvious to one having ordinary skill in the art and the teachings of London and Hallberg before them at the time the present invention was made, to use the business application Microsoft Excel, complete with the database functionality and querying, in the business application interface server system method disclosed by London. One would have been motivated to allow a user to use Microsoft Excel, in particular the database functionality of Excel, in order to better manage data, as suggested by Hallberg (page 383: "organizing your data").

London and Hallberg disclose method as above, but do not specifically disclose that the requests and outputs are XML documents specified by XSLT stylesheets. However, London does disclose the use of the X-Protocol to generate an output. One of ordinary skill in the art at the time the present invention was made would know to that what is written one programming language can also be written in another programming language such as C, C++, HTML, JavaScript, X-Protocol, XML, WML, etc. Therefore, it would have been obvious to one of ordinary skill in the art at the time the present invention was made to generate the output in XML format, and to specify the presentation of a class of XML documents using XSLT stylesheets. One would have been motivated to do this in order to have a result more compatible with web-based applications.

Claims 11 and 12: London discloses a method in a server system for providing information relating to a business application, the method comprising:

providing a default format for output of the business application (col 6 lines 63-65);

receiving from a client system a request to execute a command of a business application (col 4 lines 48-53), the command optionally indicating a user agent format or a client-specified format for the output of the business application (col 6 lines 56-58);

generating a data element by executing the command of the business application to (col 6 lines 27 "character string Hello");

executing the command of the business application to generate output (col 5 lines 22-31) from a configurable subset of the data element (col 6 lines 27 "Hello" is a character string, i.e. a configurable subset of the data element); and

sending the generated the output in the selected format to the client system (col 5 lines 22-31).

While London does not specifically disclose that the command comprises an execute, command, and argument element and further that the command element is a predefined query, London does disclose the use of "MICROSOFT EXCEL" (col 2 line 62). Hallberg discloses that Microsoft Excel is a spreadsheet program with database functionality (page 383: "Building Excel Databases"). This functionality includes the ability to selectively filter list results (page 394: "AutoFilter"), i.e. query the database ("*command element*" to return ("*execute element*") results based on a predefined filter ("*argument element*"). Therefore, it would have been obvious to one having ordinary skill in the art and the teachings of London and Hallberg before them at the time the present invention was made, to use the business application Microsoft Excel, complete with the database functionality and querying, in the business application interface server system method disclosed by London. One would have been motivated to allow a user to use Microsoft Excel, in particular the database functionality of Excel, in order to better manage data, as suggested by Hallberg (page 383: "organizing your data").

London and Hallberg disclose method as above, but do not specifically disclose that the requests and outputs are XML documents specified by XSLT stylesheets. However, London does disclose the use of the X-Protocol to generate an output. One of ordinary skill in the art at the time the present invention was made would know to that what is written one programming language can also be written in another programming language such as C, C++, HTML, JavaScript, X-Protocol, XML, WML, etc. Therefore, it would have been obvious to one of ordinary skill in the art at the time the present invention was made to generate the output in XML format, and to specify the presentation of a class of XML documents using XSLT stylesheets. One would have been motivated to do this in order to have a result more compatible with web-based applications.

Claims 2 and 26: London and Hallberg disclose the remote business application server system method as in claims 1 and 26 above respectively, and London further discloses wherein the argument element of the command indicates a type of user interface element to return (col 4 lines 5-6).

Claims 3 and 27: London and Hallberg disclose the remote business application server system method as in claims 1 and 26 above respectively, but do not specifically disclose that the argument element of the command indicates a type of user interface element to not return. However, London does disclose the request indicating a type of user interface element to return (col 4 lines 5-6). Therefore, it would have been obvious to one of ordinary skill in the art at the time the present invention was made that if one could return types of elements, one could also not return those elements. One would have been motivated to not return these elements in order to provide the server system with only specific elements, thereby limiting the amount of processes required and raising the efficiency of the processor.

Claims 4 and 28: London and Hallberg disclose the remote business application server system method as in claims 3 and 27 above respectively, but do not specifically disclose that the type of user interface element to not return is navigation data. However, one of ordinary skill in the art at the time the present invention was made would know that user interface elements of typical applications include menu bars, toolbars, backgrounds, colors, forms, shapes, navigational information, etc. Therefore, it would have been obvious one of ordinary skill in the art at the time the present invention was made to not return one of these elements. One would have been motivated to not return these elements in order to provide the server system with only specific elements, thereby limiting the amount of processes required and raising the efficiency of the processor.

Claims 5 and 29: London and Hallberg disclose the remote business application server system method as in claims 1 and 25 above respectively, and London further discloses wherein the argument element of the command comprises an "SWEDataOnly" argument, that is, when this argument is TRUE only data elements are returned and when this argument is FALSE both data and user interface elements are returned (col 5 lines 40-60).

Claims 6 and 30: London and Hallberg disclose the remote business application server system method as in claims 1 and 25 above respectively, and London further discloses wherein the argument element of the command includes a "SWEExclude" argument (col 5 lines 40-60).

Claims 7 and 31: London and Hallberg disclose the remote business application server system method as in claims 1 and 25 above respectively, and Hallberg further discloses the ability of Microsoft Excel to selectively filter database information using predefined queries (page 394). Therefore, it would have been obvious to one having ordinary skill in the art and the teachings of London and Hallberg before them at the time the present invention was made, to receive a predefined query, as taught by Hallberg, in the business application method of London and Hallberg. One would have been motivated in order to provide a user with information pertinent to them, as suggested by Hallberg (page 394 paragraph 1).

Claim 10: London and Hallberg disclose the remote business application server system method as in claim 8 above, and London further discloses wherein the argument element of the command includes a "SWEXslStyleSheet" argument (col 5 lines 40-60).

Claim 13: London and Hallberg disclose the remote business application server system method as in claim 11 above, and London further discloses the user-agent format being based on a type of user agent specified in the request (col 2 lines 60-64).

Claim 14: London and Hallberg disclose the remote business application server system method as in claim 13 above, but do not specifically disclose that the type of user agent specified is a type of browser. However, London discloses the host application that the user wishes to use being "MICROSOFT EXCEL" or "WORD FOR WINDOWS" (col 2 lines 60-64). Therefore, it would have been obvious to one of ordinary skill in the art at the time the present invention was made that London's list of host applications could be expanded to include programs such as "MICROSOFT POWERPOINT" or "MICROSOFT INTERNET EXPLORER", the later being a web browser. One would be motivated to include these in order to provide the remote user with more options for applications to run in their native GUI system.

Claims 15-18: London and Hallberg disclose the remote business application server system method as in claim 11 above, but do not specifically disclose the formats being a markup language. However, London does disclose the use of the X-Protocol to generate an output (col 5 line 25-26). One of ordinary skill in the art at the time the present invention was made would know that what is written one programming language can also be written in another programming language such as C, C++, HTML, JavaScript, X-Protocol, XML, WML, etc. One would have been motivated to use a markup language in order to have a result more compatible with web-based applications.

Claim 19: London and Hallberg disclose the remote business application server system method as in claim 11 above, but do not specifically disclose the request including a "SWESetMarkup" argument that specifies the client-specified format as being XML, HTML, or WML. However, London does disclose the use of the X-Protocol to generate an output (col 5 line 25-26) as well disclosing the host application that the client wishes to use being "MICROSOFT EXCEL" or "WORD FOR WINDOWS" (col 2 lines 60-64). One of ordinary skill in the art at the time the present invention was made would realize that London's list of host applications could be expanded to include programs such as "MICROSOFT POWERPOINT" or "MICROSOFT INTERNET EXPLORER", the later being a web browser. In the case of "MICROSOFT INTERNET EXPLORER" being the application the client wishes to use, it would have been obvious to one of ordinary skill in the art at the time of the present invention that the original programming could be written in a more web-friendly language such as HTML, XML, WML, or JavaScript. One would be motivated to do this in order to provide the client system with more selection in applications to use, as well as providing a result that is more compatible in the case of web-based applications.

9. **Claim 24** is rejected under 35 U.S.C. 103(a) as being unpatentable over Bookman in view of Hallberg.

Claim 24: Bookman discloses a computer-readable medium containing: instructions executable on a computer system, configured to execute a command of a business application; and
a data structure defining the command, wherein the command is outbound to a web server and the web server is configured to execute on the computer system (col 3 lines 61-62: "Web server environment, containing conventional objects", col 4 lines 12-13: "back to requesting Web browser"), the data structure comprising:

an application element (col 1 lines 56-57: "CGI is a standard interface for running external programs on a Web server.", col 3 lines 62-63: "an object") having a name attribute (col 3 lines 65-66: "each object has an attribute associated with it that identifies the type of object");

a navigation element nested within the application element, having a name attribute (col 4 lines 6: "based on the object request URL").

Bookman does not specifically disclose that the navigation element has sub-elements from a set comprising a menu element, tool bar element, screen bar element, thread bar element, view bar element, and page item element. Bookman does disclose "Web browsers" (col 1 lines 36-44). It would have been obvious to one of ordinary skill in the art at the time the present invention was made that "Web browsers" include: menus, tool bars, screen bars, view bars, etc. One would have been motivated to disclose the navigation element having these standard browser sub-elements in order to allow the user greater flexibility in the browser they chose to use.

Also, Bookman do not specifically disclose one or more elements from the set of elements including a screen element, an applet element, and a form element, the one or more elements being nested within the application element and each having a name attribute. However, Bookman do disclose Web browsers requesting particular hypermedia documents (col 1 lines 45-67, col 2 lines 1-8). Therefore, it would have been obvious to one of ordinary skill in the art at the time the present invention was made to allow applet scripts, forms, and URL queries to be nested within the web browser. One would have been motivated to do this in order to provide dynamic hypermedia to an end-user, thereby increasing the user interactivity.

Bookman does not specifically disclose that the data structure is an XML document. However, Bookman do disclose the use of object requests from Web server executables (col 4 lines 2-3) and the use of HTML (col 4 lines 45-60). Therefore, it would have been obvious to one of ordinary skill in the art at the time the present invention was made that the data structure to be used in a Web based situation could also be written in XML, WML, HTML, JavaScript, CGI Script, etc. One would be motivated to write it in XML in order to benefit from a more database oriented programming language.

Bookman does not specifically disclose a predefined query bar element nested within the application element. Hallberg discloses that Microsoft Excel is a spreadsheet program with database functionality (page 383: "Building Excel Databases"). This functionality includes the ability to selectively filter list results, i.e. query the database to return results based on a predefined filter, used in bars (page 394: "AutoFilter", Fig. 14.9). Further, Hallberg discloses that Microsoft Excel files can be opened using a web browser (page 638: "The file will open within Internet Explorer, and you can edit it just as if you had opened it in Excel directly.") Therefore, it would have been obvious to one having ordinary skill in the art and the teachings of Bookman and Hallberg before them at the time the present invention was made, to use the business application Microsoft Excel, complete with the database functionality and predetermined query bars, in the business application interface server system disclosed by Bookman. One would have been motivated to allow a user to use Microsoft Excel while in a web browser, in particular the database functionality of Excel, in order to better manage data, as suggested by Hallberg (page 383: "organizing your data").

Bookman further discloses that the web server environment contains a multitude of computers (col 1 lines 12-14) and programs (col 1 lines 22-24). Data is supplied to and from the computers and servers (col 1 lines 44-52). Therefore it would have been obvious to one having ordinary skill in the art and the teachings of Bookman and Hallberg before them at the time the present invention was made to apply the known technique of a network of computers and a server and passing of information between them, taught by Bookman, improve the similar device suggested by Bookman and Hallberg above to yield the predictable result of expanding Bookman and Hallberg's device above to further contain a plurality of instructions executable on a plurality of computer systems and configured to execute a plurality of commands on a plurality of business applications according to the data structures and methods of Bookman and Hallberg above.

Response to Arguments

10. Applicant's arguments filed October 15, 2008 have been fully considered but they are not persuasive.

Applicant argues, on page 10, that the amendments made to Claims 20 and 22-24 overcome the non-statutory 35 U.S.C. 101 rejection. The Examiner respectfully disagrees. As shown in the rejection above, "computer-readable storage medium" does not overcome the non-statutory rejection because the specification still provides for the "storage" to be a propagation medium. A propagation medium is not a statutory category of invention and the rejection has been maintained.

Applicant argues, on pages 10-11, that the amendments made to claims 20 and 22-24 overcome the previous anticipatory rejection of Bookman. The Examiner notes the new grounds of rejection presented above. Further, Bookman does suggest a plurality of computer systems in at least the background of the art. It is therefore obvious to predictably implement the method Bookman, and in the case of claim 24 Bookman and Hallberg, in a multi-computer environment. The argument is not persuasive.

Applicant argues, on pages 11-12, that the amendments made to claims 1, 8 and 11 overcome the obviousness rejection previously presented. Particularly, Applicant argues that neither London nor Hallberg teach or suggest the generation of a UI element from a configurable subset of data. The Examiner respectfully disagrees. As shown in the respective rejections above, London discloses the use of a character string data element to generate the UI element. Character strings are configurable data sets. A subset of the character string data set, "Hello", is used in the particular embodiment disclosed by London. The argument is not persuasive.

Applicant argues, on page 13, that claim 25 is not unpatentable over Bookman. Particularly, Applicant argues that London does not disclose otherwise sending a second response that does not include the user interface element. Bookman clearly discloses this limitation (col 5 lines 60-67), where the UI element is passed optionally (first request) otherwise it is not passed. The argument is not persuasive.

Conclusion

11. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Andrew Tank whose telephone number is 571-270-1692. The examiner can normally be reached on Mon - Thur 0830-1700 EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, William Bashore can be reached on 571-272-4088. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/A. T./
Examiner, Art Unit 2175
January 31, 2009

/WILLIAM L. BASHORE/
Supervisory Patent Examiner, Art Unit 2175